

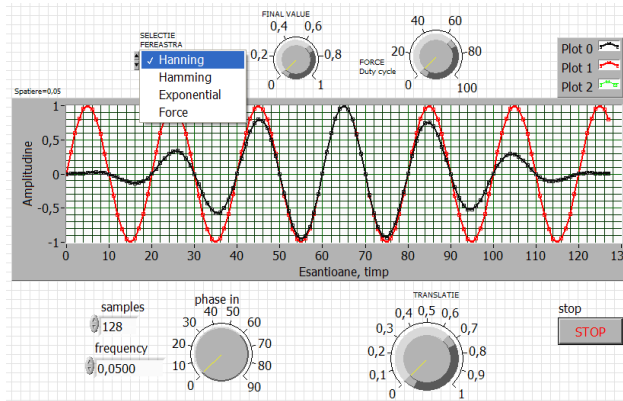
LABVIEW + cod Matlab și C - semnale (1D) ponderate (windowing)

Prof. Iulian Lupea, UTCluj

1. Se generează un semnal sinusoidal discret apelând Sine Wave.vi (folosiți controale în PF pentru intrările: samples= n , frequency, phase in). Semnalul poate fi prelucrat prin una din cele patru ferestre de ponderare predefinede: Hanning, Hamming, Exponential și Force (paleta Signal Processing/ Windows).

Fiecare eșantion din semnal este înmulțit (ponderat) cu câte un coeficient specific ferestrei (total n înmulțiri); coeficienții unei ferestre sunt calculați prin relații de calcul specifice ferestrei.

În aplicație, semnalul este prelucrat automat de o fereastră selectată de operator și în paralel același semnal este înmulțit (prin program) cu coeficienți generați după relația de calcul, rezultând două semnale identice suprapuse.

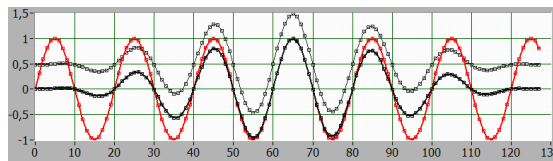


2. Prelucrare tablou prin fereastră Hanning - Matlab

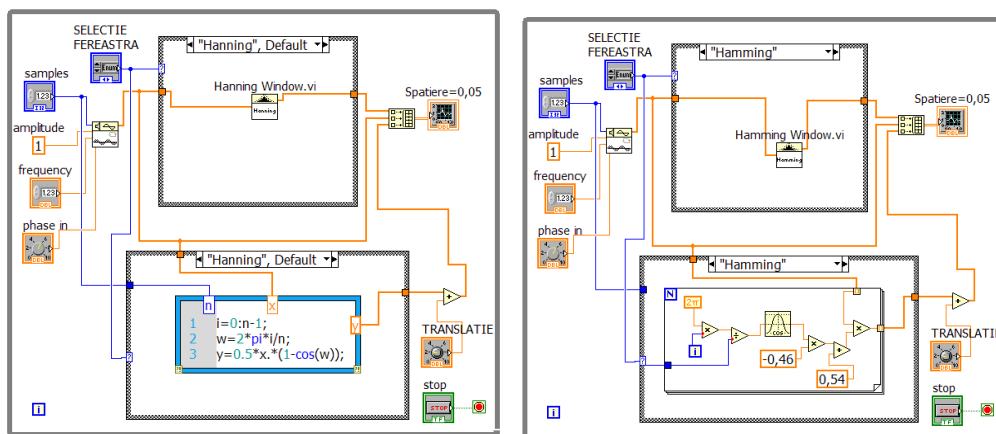
În figura 1 observăm curba sinus inițială de amplitudine constantă și două curbe identice suprapuse, prelucrate de fereastră **Hanning** (cu amplitudine zero la început și la sfârșit) și respectiv curba obținută separat prin formula de calcul.

$$y_i = 0.5x_i[1 - \cos(w)]$$

$$w = \frac{2\pi i}{n} \quad i = 0, 1, 2, \dots, n-1,$$



Controlul TRANSLATIE permite separarea celor două curbe identice (Fig. 2). Prin aceasta se verifică corectitudinea programării explicite asociate ferestrei predefinede.



Ponderarea cu coeficienții ferestrei Hanning se face într-un caz separat (Fig. 3) printr-un segment de cod/script scris în Matlab (trebuie să existe produsul Matlab instalat în același calculator). y_i sunt valorile semnalului rezultat iar x_i sunt eșantioanele semnalului sinus inițial. *Rescrieți formulele Hanning în Labview sau cod C.*

În aplicația generală sunt folosite două instrucțiuni Case (4 cazuri în fiecare) și un selector comun Selectie Fereastră de tip Enum. Build Array unește 3 semnale tablou/array 1D într-un tablou/array 2D.

3. Fereastră Hamming - cod Labview

Ponderarea cu fereastră Hamming se face în paralel într-un alt caz folosind programare Labview. y_0 și y_{n-1} sunt val. minime.

Fereastra Hanning anulează prima valoare din semnal iar Hamming n-o anulează; modificați faze în pentru a porni semnalul sinusoidal de la o valoare nenulă și a observa comparativ ferestrele.

$$y_i = x_i[0.54 - 0.46\cos(w)]$$

$$w = \frac{2\pi i}{n} \quad i = 0, 1, 2, \dots, n-1$$

4. Fereastra Exponential - cod C în formula Node

La ponderarea cu fereastra Exponentială se apelează la formula nod folosind codul limbajului C. Final value (f) este valoarea finală minimă, controlată din panoul frontal în limitele 0..1.

$$y_i = x_i \exp(a \cdot i)$$

$$a = \frac{\ln(f)}{n-1} \quad i = 0, 1, 2, \dots, n-1$$

Modificați controlul Final Value asociat ferestrei Exponential și observați efectul.

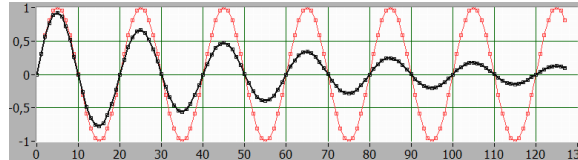


Fig. Final value =0.12

5. Fereastra Force - cod Labview

La ponderarea cu fereastra Force se apelează la programare în Labview. Controlul Duty cycle% este în procente 0%.. 100%.

$$w = \begin{cases} x_i & \text{(if } 0 \leq i \leq d) \\ 0 & \text{elsewhere} \end{cases}$$

$$d = (0.01)(n)(\text{duty cycle})$$

Modificați controlul Duty cycle asociat ferestrei Force și observați efectul.

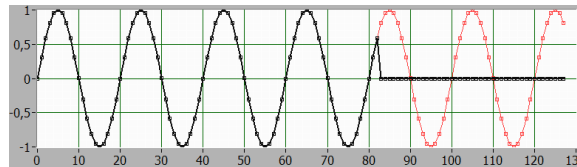
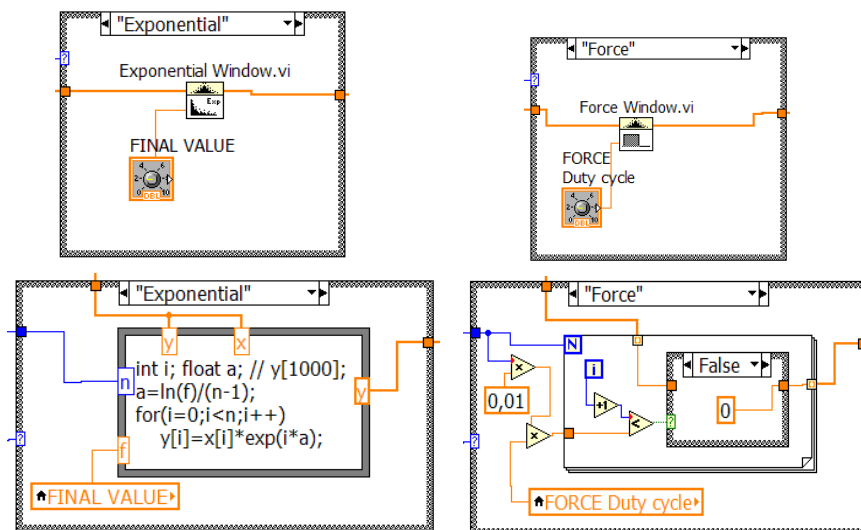


Fig. Force duty cycle= 65%



6. Aplicații

6.1. Introduceți posibilitatea de a selecta dintr-o listă semnalul de prelucrat. Lista conține trei semnale Sine wave.vi, Square wave.vi și un semnal aleator.

6.2. Folosiți o singură instrucțiune CASE în aplicație unificând cazurile omoloage; astfel fereastra predefinită și relațiile de calcul explicite asociate se vor regăsi în același caz din instrucțiunea CASE.

6.3. Adăugați noi ferestre de ponderare (fereastra Blackman - cazul 5, etc.)

Obs: Prelucrarea semnalului cu ferestre de pondere (Windowing) → realizează diminuarea 'spectral leakage'.